

Software de control de movimiento de plataforma motorizada a dos ejes para el micromaquinado con láser de imágenes 2D

Nonaka M., Kovalsky M., Agüero M., Hnilo A.

Centro de Investigaciones en Láseres y Aplicaciones (CEILAP), UNIDEF (MINDEF - CONICET)

J.B. de la Salle 4397 (B1603ALO) Villa Martelli, Pcia. de Buenos Aires - Argentina

mnonaka@citedef.gob.ar , mkovalsky@citedef.gob.ar

Resumen- Se desarrolló un software con una interfaz gráfica fácil de utilizar que permite el control mediante computadora del movimiento en dos direcciones de una plataforma que utiliza motores paso a paso con una resolución de 0.1 μm . Además, en conjunto con el desplazamiento, comanda en forma automática el

encendido y apagado de un láser de Nd:YAG con Q-Switch activo logrando así generar imágenes micromaquinadas sobre diferentes sustratos.

control por motores por pasos sobre el cual se coloca la muestra que se desea micromaquinar.

I. INTRODUCCIÓN

Los láseres han sido empleados ampliamente en maquinado y corte de metales desde principios de la década del 1970 [1]. Sin embargo, es con el desarrollo de los láseres de estado totalmente sólido, que su aplicación para maquinar materiales no metálicos como cerámicos, plásticos y semiconductores (en particular silicio) cobró gran importancia, especialmente en las industrias electrónica y opto-electrónica y en el desarrollo de prototipos de laboratorio [2,3]. La capacidad de los láseres pulsados de maquinar con precisión del orden del micrón materiales difícilmente maquinables por otros medios y la posibilidad de diseñar piezas únicas sin las costosas técnicas de litografía, contribuyeron a la expansión del empleo de este tipo de láseres en el diseño y desarrollo de sistemas micro electro mecánicos (MEMS)[4] . Esto creó un interés creciente por el estudio de los parámetros que determinan la calidad del micromaquinado [5]. Un gran número de trabajos han sido publicados acerca del proceso de micromaquinado, cubriendo aspectos como la influencia de la longitud de onda, duración de pulso y frecuencia de repetición [1, 5-9] hasta los procesos físicos de interacción radiación láser – material [10,11].

Por otra parte, la importancia de desarrollar un software de control propio, que para su uso no le requiera al usuario conocimiento previo de programación, resulta esencial a la hora de realizar ensayos de laboratorio. En el presente trabajo se describe el diseño y la implementación de un software en una estación de micromaquinado con láser. La interfaz gráfica del programa permite al usuario controlar, de manera simple, diferentes parámetros durante el proceso de micromaquinado. Entre ellos se encuentra la carga automática de un archivo de imagen monocromática, incluyendo el procesamiento de sus datos y ajuste de sus dimensiones físicas. Además, este software permite controlar el tiempo de exposición de la muestra a la radiación láser, así como también el movimiento de un sistema de posicionamiento x-y (plataforma x-y) con

II. CONFORMACIÓN DE LA IMAGEN SOBRE UN SUSTRATO

Para el diseño y desarrollo del software debe tenerse en cuenta las diferentes etapas para maquinar un sustrato. La muestra se monta sobre una plataforma x-y con control de movimiento. En la Fig. 1 se presenta un esquema de la secuencia de pasos a considerar. En particular, se describe el micromaquinado con láser de dos líneas rectas separadas por una cierta longitud. Para su realización se requiere:

- A. Colocar la muestra sobre la plataforma (4). Luego, activar el láser. El haz (1) se enfoca sobre la superficie de la muestra con una lente convergente ($f = 25 \text{ mm}$).
- B. Mover la plataforma motorizada en línea recta (dirección x en la figura) una cierta longitud. Al estar encendido el láser, a medida que la plataforma se desplaza, el haz impacta sobre la muestra y realiza una marca sobre el sustrato. Notar que la óptica se encuentra fija a diferencia de la plataforma, que sí es móvil.
- C. Apagar el láser.
- D. Mover la plataforma sobre la misma línea del paso B. Como el láser no se encuentra activado, al desplazarse, no se produce marca alguna sobre el sustrato. Se posiciona la plataforma en el punto inicial de la segunda línea.
- E. Se procede a realizar el siguiente tramo de línea. Para ello, se vuelve a activar el láser, y se repiten los pasos B y C.

Observar que a partir de una combinación de estos desplazamientos, con la activación y/o desactivación del láser, es posible micromaquinar con láser diseños complejos a partir de patrones sencillos.

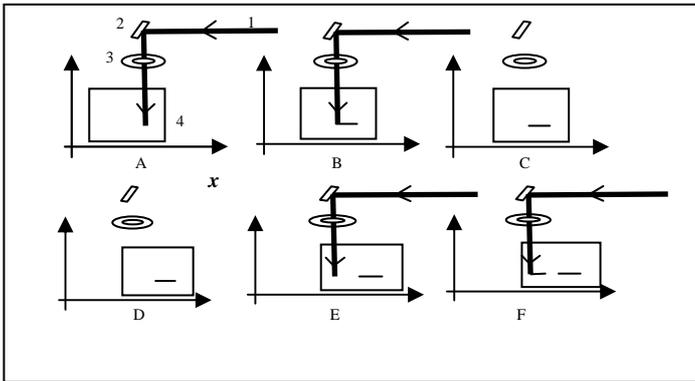


Fig. 1. Metodología de conformación de imágenes. 1: haz del láser; 2: espejo HR 45°; 3: óptica de enfoque; 4: plataforma x-y motorizada. Secuencia de pasos: A, B, C, D, E, F (equivalente a B).

En las Fig. 2 y 3 se muestran la disposición real de los elementos utilizados en el laboratorio para efectuar los ensayos. En la publicación "Imágenes 2D de alta resolución micromaquinadas con láseres de nanosegundos" [12] se detallan los aspectos físicos de la cavidad láser y el sistema óptico utilizado.

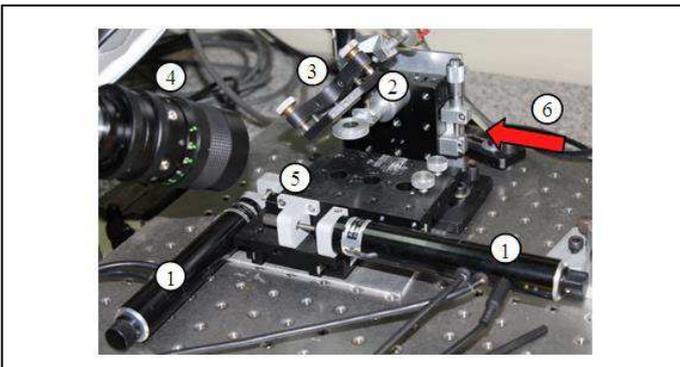


Fig.2. Montaje de la plataforma x-y. 1: Tornillos micrométricos; 2: Lente de enfoque; 3: Espejo deflector a 45°; 4: Cámara CCD; 5: Base porta muestra; 6: Láser de escritura.

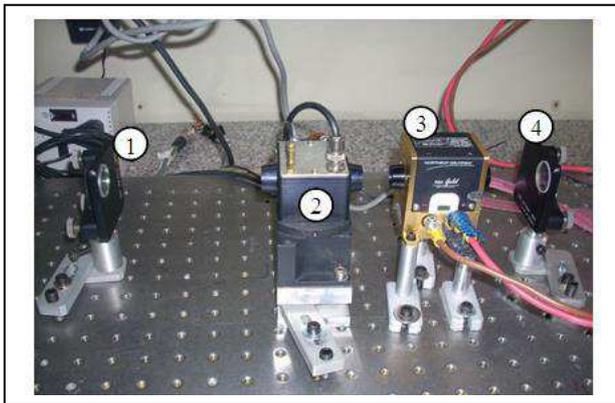


Fig. 3. Láser de Nd:YAG. 1: Espejo con radio de curvatura de 1m y R=80%; 2: Modulador acusto-óptico; 3: Cabezal láser; 4: Espejo plano HR.

III. PROCESAMIENTO DE DATOS DE LA IMAGEN

El proceso de micromaquinado comienza con la lectura de la imagen que se desea maquinar sobre el sustrato. El software interpreta estos dibujos y los convierte en trayectorias de movimiento de la muestra. El software diseñado lee archivos de imagen en formato .bmp monocromo. Dicha extensión facilita detectar píxeles en blanco y negro. En la Fig. 4 (1) se puede visualizar una imagen monocromo de un código QR. La misma, traducida en datos hexadecimales, tiene la estructura que se muestra en la Fig. 4 (2).

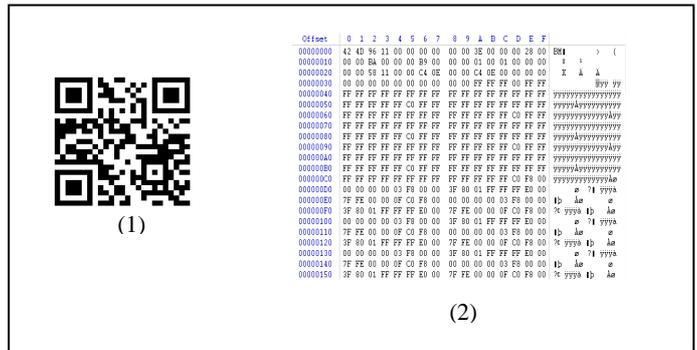


Fig. 4. Imagen de un código QR. (1) en formato .bmp monocromo; (2) en un archivo hexadecimal.

Para poder interpretar los datos dentro del archivo en formato en hexadecimal, se deben identificar sus diferentes campos constitutivos (ver Fig. 5). Para el caso de un archivo .bmp monocromo, la secuencia de bytes comienza con las letras B M en ASCII, expresadas en hexadecimal como 42h 4Dh, seguida por 4 bytes que expresan el tamaño del archivo en bytes (en donde el primer byte es el menos significativo).

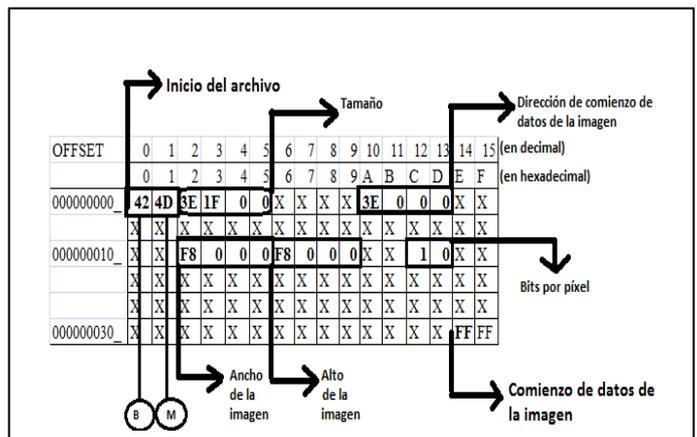


Fig. 5. Descripciones del contenido del archivo de la imagen monocromo en formato hexadecimal.

Por ejemplo, si se tiene 3Eh 1Fh 00h 00h el tamaño del archivo será 00h 00h 1Fh 3Eh, es decir, 7998 bytes. La información sobre el alto y el ancho de la imagen se localizan en las direcciones 12h y 16h, respectivamente. Para detectar el comienzo de los datos del dibujo en monocromo se deben leer

los 4 bytes contenidos dentro del décimo al décimo tercer byte respecto del inicio del archivo. Si se tiene un comienzo en 3Eh 00h 00h 00h, significará que los datos del dibujo monocromo empezarán en la dirección 3Eh, o sea, 62 bytes a partir del origen del archivo. Se debe destacar que dicho valor de comienzo de datos corresponde al último byte que describe a la imagen (los bytes de un archivo monocromo se encuentran dispuestos de atrás para adelante, es decir, el primer byte del comienzo de los datos de la imagen corresponde a los 8 últimos píxeles de la imagen monocromo).

Como se ha dicho con anterioridad, a partir de la dirección en donde comienzan los datos de la imagen (en el ejemplo anterior, la dirección 3Eh respecto del origen), se pueden visualizar sucesivos bytes expresados en hexadecimal que se refieren a los píxeles de la imagen monocromo. Por ejemplo, en la dirección de comienzo de los datos de la imagen, corresponde a los últimos 8 píxeles de la imagen que se muestran en la Fig. 4 (1). A modo ilustrativo, si se supone que en dicha dirección se tiene un valor en hexadecimal de FFh, es decir, 1111 1111 en binario (en donde el 1 implica que el píxel es de color blanco), se tendrán los últimos 8 píxeles de la imagen de color blanco. En cambio, si en vez de FFh se tiene un valor de A5h, 1010 0101 en binario (donde 0 implica que el píxel es de color negro), para este caso, el último píxel es de color negro, el anteúltimo de color blanco, etc.

En resumen, la idea de conformación de imágenes es la de barrer, a partir de la dirección de comienzo hasta la de fin de datos de la imagen, los distintos bytes en hexadecimal correspondientes a la imagen monocromo, obteniendo así los colores en blanco y en negro de cada píxel en particular. Para ello, se desarrolló un algoritmo mediante el cual se puede acceder a los distintos campos del archivo hexadecimal (tamaño del archivo, alto y ancho de la imagen, y dirección de comienzo de datos de la imagen, entre otros). Además, este algoritmo permite procesar cada byte identificando píxeles en color blanco y negro, como así también la posición de cada píxel de la imagen (es decir, detectando si se trata del último píxel de la imagen, anteúltimo píxel de la imagen, etc.).

IV. DIMENSIONAMIENTO Y AJUSTES

Una vez obtenidos los colores de cada píxel, que en conjunto permiten conformar una imagen monocromo, se realizó el dimensionamiento de la imagen. Esto significa, cuánto se debe desplazar la plataforma x-y por cada píxel para poder lograr una imagen del tamaño deseado. Aquí se aplica el criterio visto en la sección II: los píxeles en color negro serán análogos al efecto de desplazar los motores al mismo tiempo que el láser se encuentra encendido (micromaquinar). Por el contrario, los píxeles en color blanco serán análogos al efecto de desplazar los motores mientras el láser se encuentre apagado (no micromaquinar).

El estudio del desempeño del software se realizó utilizando una plataforma de desplazamiento x-y motorizada Zaber de la

serie T-LA con las siguientes especificaciones: rango de 60mm; resolución de 0,1µm y repetibilidad <0,4µm.

Se halló la correspondencia entre el microstep del motor y una longitud. Por defecto, el dispositivo define 64microstep por paso. Es decir, si se tiene una plataforma T-LA que se mueve a 304.8µm por revolución del motor, y se conoce la especificación de los números de pasos por revolución (48 pasos en este caso) del motor, se obtiene la distancia que se desplaza la plataforma por microstep (1 microstep \equiv 0.09921875 µm).

Por ejemplo, si se quiere un desplazamiento de 60mm, el motor deberá desplazarse 606208 microstep. Con este criterio, si se desea realizar una imagen de un cierto ancho en mm, sabiendo la cantidad de píxeles en dicha longitud, se puede traducir esa longitud en cantidad de microsteps.

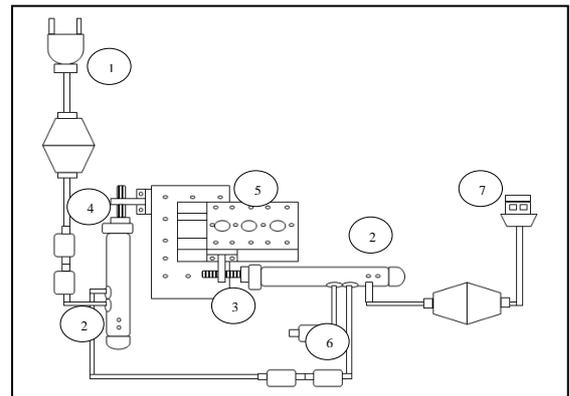


Fig. 4. Esquema del posicionador Zaber de la serie T-LA. 1: alimentación; 2: leds de indicación; 3: motor 1; 4: motor 2; 5: plataforma x-y; 6: entrada al dispositivo que enciende y apaga al láser; 7: comunicación con la PC.

Se desarrolló un algoritmo que habilita al usuario a cargar una imagen e ingresar por teclado el valor del tamaño deseado, realizando automáticamente las conversiones pertinentes de escala que se explicó en el párrafo anterior.

V. PROTOCOLO DE COMUNICACIÓN Y CONTROL DEL MOVIMIENTO

El plataforma x-y posee un protocolo de comunicación RS232 (9600 baudios, sin handshaking, 8 bits de datos, sin paridad y un bit de stop). El formato de instrucción se compone por 6 bytes: el byte 1 refiere al dispositivo (si es el motor que controla el eje de las abscisas o si el que controla el de las ordenadas); el byte 2 refiere al comando a aplicar (desplazar a una posición absoluta, relativa, etc.); del byte 3 al 6, corresponden al dato de 4 bytes en donde el byte menos significativo se guardará en el byte 3 y el más significativo en el byte 6.

En la Fig. 5 se observa el esquema básico de comunicación. El programa, mediante la PC, se comunica con la plataforma x-y en forma bidireccional por medio de datos seriales, sincronizando el encendido del láser con el desplazamiento de la plataforma sobre la cual se coloca el sustrato.

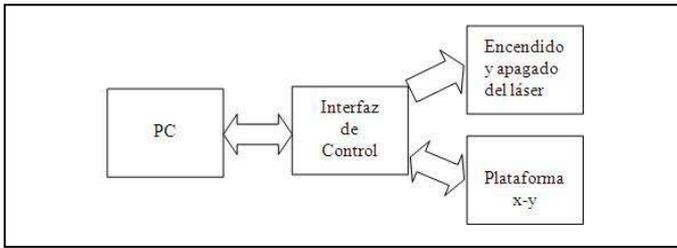


Fig. 5. Esquema básico de comunicación.

Los datos que se transmiten deben respetar el formato de instrucción, es decir, indicar a qué motor se le está indicando el comando a operar, qué tipo de comando debe ejecutar dicho motor, y en qué magnitud. Por ejemplo, si se quiere desplazar el motor 1 unos 60 mm desde el origen de coordenadas, se le debe transmitir los siguientes 6 bytes: el byte 1, 01h; el byte 2, 42h (comando de desplazamiento absoluto); del byte 3 al 6, 00h 04h 09h 00h.

La interfaz de control opera a través de un microcontrolador que habilita el disparo del modulador acústico-óptico del láser, funcionando como un obturador de alta velocidad.

A continuación, se detallan los pasos que permiten la automatización del movimiento:

- 1) Se selecciona una imagen .bmp monocromo.
- 2) El software analiza los datos de la imagen. Si bien, el estudio de un archivo .bmp es un tema conocido, lo novedoso es poder, a partir del análisis de píxeles, controlar el encendido y apagado de un láser (que opera a mil pulsos por segundo). Para ello, es esencial conocer cuándo hay un píxel en blanco (láser apagado) y cuándo uno en negro (láser encendido).

Al leer un píxel de color negro, el software le envía un comando a la interfaz de control vía serie. Esta interfaz está compuesta por un PIC que se comunica con el modulador de frecuencia que realiza el Q-switch de la fuente del láser. Es decir, activa el láser con la frecuencia que se ingresa en el software, en este caso a 1kHz. Para su desactivación, espera por la lectura de un píxel de color blanco y luego, se establece al modulador de frecuencia que controla la fuente a 0Hz.

- 3) Para realizar el desplazamiento de los motores, se efectúa un barrido de datos por cada línea que compone la imagen. En la Fig.6 se muestra una imagen .bmp de 180 píxeles por 14 píxeles, es decir, se debe barrer 14 líneas para conformar la imagen. En la primera línea, los 180 píxeles son de color blanco, por lo cual, se le enviará una instrucción al PIC para que opere a 0Hz (es decir, que no se module a la fuente del láser). Al mismo tiempo, vía serie, se le consignará a los motores la siguiente instrucción contenida en 6 bytes: el número de motor a mover (0x01 para el motor en el eje de las abscisas y 0x02 para el motor en el eje de las ordenadas); un comando que indica la acción a ejecutar, 0x14 para mostrar que se trata de un desplazamiento absoluto y 4 bytes consecutivos

ordenados como little-endian, que denotan la posición absoluta a la cual desplazarse en unidades de microstep. Para determinar la longitud que debe trasladarse, se cuenta el número de píxeles de color blanco o negro que hay en la imagen y se lo multiplica por la cantidad de mm a la cual equivale cada píxel (si se quiere una imagen de 10mm y la misma posee 180 píxeles de longitud, cada píxel tendrá un equivalente a 180 píxeles dividido 10mm).

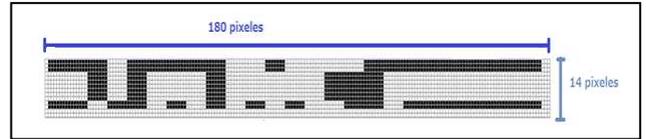


Fig. 6. Imagen .bmp monocromo de 180 píxeles por 14 píxeles.

Para maquinar la línea siguiente, se debe mover el motor que controla al eje de las ordenadas un píxel hacia abajo (el barrido de datos de la imagen se ejecuta de izquierda a derecha y de arriba hacia abajo). Se realiza el mismo mecanismo de movimiento: se comienza a barrer los datos de una línea, se cuenta el número de píxeles de color negro o blanco (en el caso de ser píxeles negros, se envía al PIC el dato para que el modulador de frecuencia active el láser) y se realiza el desplazamiento en conjunto. En el caso de tener píxeles en blanco, se envía al PIC el dato para no activar la fuente del láser y, en conjunto, realiza el desplazamiento. La imagen se conforma, de esta forma, como lo explicado en el apartado II.

VI. SOFTWARE DE CONTROL

El software desarrollado (Fig. 7) es una herramienta indispensable para tener control del proceso de acuerdo con las necesidades del usuario. A continuación se describen sus características principales. Permite el micromaquinado con láser de cualquier imagen monocromática (extensión de archivo .bmp monocromo); habilita al usuario a dimensionar la imagen monocromática a escalas reales deseadas; posee una interfaz gráfica fácil de manipular por cualquier usuario; salvaguarda

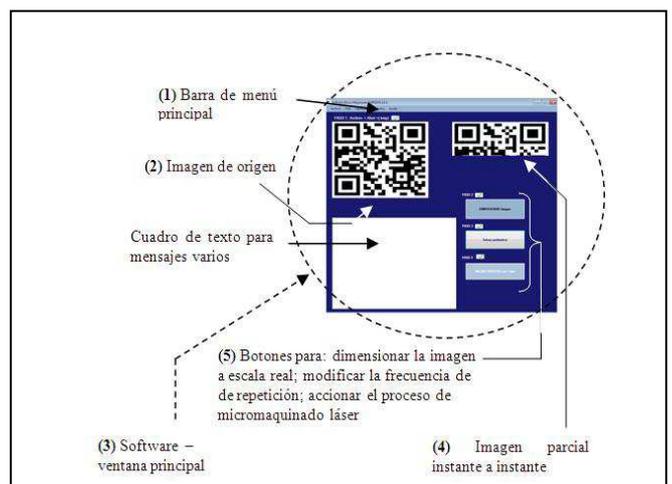


Fig. 7. Captura en pantalla de la página principal del Software de micromaquinado.

en formato de texto los parámetros bajo los cuales se realizó el ensayo como así también la fecha y hora de realización. Esto es de esencial importancia en las experiencias de laboratorio, permitiendo al usuario la visualización en pantalla del proceso de micromaquinado de la muestra en tiempo real. Además, posibilita que el usuario realice pruebas previas antes de comenzar el proceso (testeo de la ubicación de la muestra, enfoque del láser, etc.). El software se desarrolló en lenguaje de programación C# y opera bajo el Sistema Operativo Windows.

VII. RESULTADOS. IMÁGENES MICROMAQUINADAS CON LÁSER

El estudio de las prestaciones del software se realizó micromaquinando con láser obleas de silicio de 350 µm de espesor. En las Fig. 8, 9 y 10 se muestran diferentes diseños obtenidos, operando el láser en diferentes condiciones de funcionamiento. La integración del software en la estación de micromaquinado permitió la ejecución exitosa de diversos micromecanizados 2D.



Fig. 8. Imagen de un oso panda realizada con micromaquinado láser. El grabado se compara con una moneda de 5 centavos (pesos argentinos).

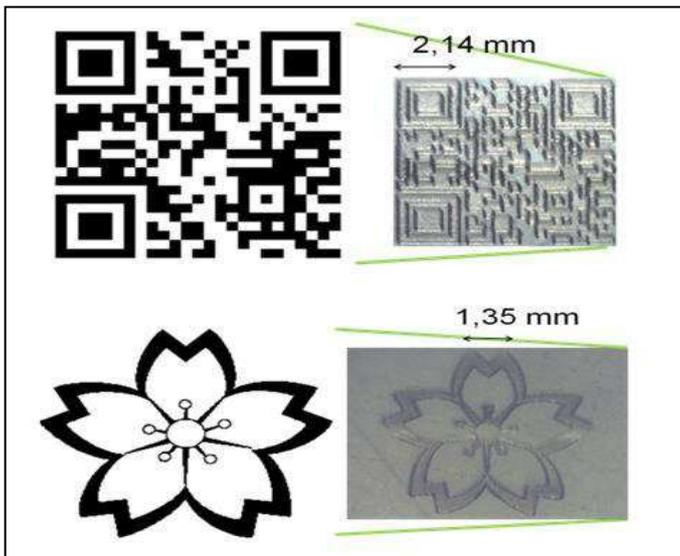


Fig. 9. Imágenes de archivo (.bmp monocromo) micromaquinadas en sustrato de Si: Código QR y flor del cerezo.

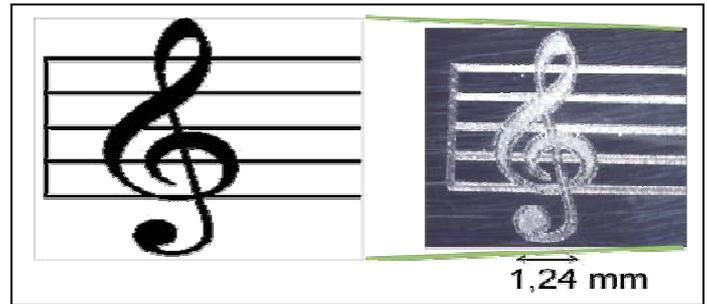


Fig. 10. "Clave de Sol", grabada por micromaquinado láser sobre lámina de Si. La superficie del sustrato fue previamente pulida.

VIII. CONCLUSIONES

En este trabajo se describieron las características del software de control de una estación de micromaquinado con láser, desarrollado a la medida de las necesidades. Este programa se destaca por su facilidad de operación e independencia, ya que puede ejecutarse en cualquier computadora operando bajo Windows y no requiere para su funcionamiento de la instalación adicional de ningún otro tipo de aplicación gráfica ni de procesamiento.

El software permite el control del movimiento de un sistema de posicionamiento x-y motorizado, y el encendido y apagado de un láser de Nd:YAG con Q-Switch activo, ejecutando de manera confiable las tareas asignadas.

Se alcanzó el objetivo del desarrollo de un software propio que permite que el usuario pueda controlar a voluntad diferentes parámetros del proceso de micromaquinado. La incorporación de este software a la estación de micromaquinado con láser fue indispensable para la realización exitosa de diferentes grabados 2D con geometrías diversas.

REFERENCIAS

- [1] S.J. Ahn, D.W. Kim, H.S. Kim, K.H. Cho, S.S. Cho, Appl. Phys. A 69 (1999) 527–530.
- [2] J.F. Ready, Industrial Applications of Lasers, Academic Press, London, 1997, p. 394.
- [3] ICALEO, Laser Micro-fabrication Session, October 2–5, 2000.
- [4] Cheng Luo, Liwei Lin, Sensors and Actuators A Vol. 97 (2002) 398–404
- [5] P. Simon, J. Ihlemann, Appl. Phys. A 63 (1996) 505–508.
- [6] S. Daur, A. Ehlert, S. Bu'ttgenbach, Sens. Actuators 76 (1999)381–385.
- [7] J. Li, G.K. Anathasuresh, J. Micromech. Microeng. 11 (2001)38–47.
- [8] J. Jandeleit, A. Horn, R. Weichenhain, E.W. Kreuz, R.Poprave, Appl. Surf. Sci. 127–129 (1998) 885–891.
- [9] H.W. Bergmann, Appl. Surf. Sci. 96–98 (1996) 287–295.
- [10] W. Xu, J. Phys. Condens. Matter 10 (1998) 6105–6120.
- [11] G. Marowsky, C.K. Rhodes, Appl. Phys. B 66 (1998) 475–478.
- [12] M. Nonaka, M. Agüero, D. Krygier, A. Hnilo, M. Kovalsky, A. Tourón, L. Fidalgo, P. Diodati, O. Radulovich, Anales AFA vol 25 Nro2 (2014) 88-90.
- [13] Manual de usuario Zaber T-LA, <http://www.zaber.com/wiki/Manuals/T-LA>